Treball de Fi de Grau/Màster

Grau en Enginyeria en Tecnologies Industrial

VULNInterpreter

Creación de un aplicativo que facilite los escaneos de puertos con Python

MEMORIA

Autor: Albert Bolumar Barrera
Director: Lluís Solano Albajes

Convocatoria: junio 2019

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Resumen

El objetivo principal de este trabajo es el de realizar una herramienta con Python que sea capaz de automatizar la ejecución de un examen Nmap en busca de vulnerabilidades y brechas en los puertos red de una Ip. Además, se pretende que el mismo genere un documento resumen con las conclusiones extraídas de los resultados del examen, que suelen alargarse hasta los varios millones de líneas. También se buscará que dicho documento tenga un lenguaje lo más entendible posible. Lo que se pretende con ello es facilitar tareas relacionadas con la ciberseguridad que suelen ser muy pesadas pero no por ello menos imprescindibles para garantizar un buen nivel de protección en nuestros equipos.

Crear una interfaz con la que cualquier usuario pueda interactuar de forma fácil y ágil con el programa también es un requerimiento importante si se pretende que el uso del software llegue a personas sin ningún tipo de conocimiento técnico.

Para realizar este trabajo se han empleado multitud de recursos y herramientas, entre los cuales se pueden destacar:

- **Nmap:** popular software ejecutable desde el terminal de Linux o el cmd de Windows que permite la ejecución de escaneados en los puertos. Además, gracias a que es un programa de código libre, incorpora una enorme variedad de scripts que facilitan mucho la tarea de darle objetivos concretos al escaneado.

- **Fichero Bash:** bash es un intérprete de comandos de consola, la Shell de los sistemas operativos basados en UNIX, es decir, el intérprete de los comandos del terminal de Linux. Estos ficheros, de extensión .sh, son usados para guiar al sistema hasta la ruta correcta de directorios dónde ejecutar las órdenes.

- **SQL:** se ha usado SQL para crear una base de datos y extraer información de ella.

- **Python:** es el lenguaje en el que está escrito tanto el programa principal como el que gestiona las peticiones y las rutas del servidor de la aplicación.

- **Html, CSS:** son los lenguajes que han sido necesarios para desarrollar las interfaces de contacto con el usuario.

SUMARIO

1. Contextualización y problemática

No es necesario ser ningún tipo de genio para percatarse de la enorme dependencia que la sociedad del siglo XXI tiene de la red como sistema de comunicaciones. Prueba de ello es que según los datos presentados en la edición del 2018 del Mobile World Congres de Barcelona por la GSMA [1], la asociación de operadoras de telecomunicaciones que lo organiza, el número de tarjetas Sim había superado por primera vez la población mundial, habiendo un total de 7.800 millones de tarjetas activas aproximadamente. Es cierto que no todas ellas tienen por qué funcionar en un smartphone, no obstante, estas cifras permiten hacerse una idea del enorme orden de magnitud del número de aplicaciones, webapps, páginas web y otras utilidades que, diariamente, son ejecutados varias decenas de millones de veces por todos estos usuarios.

En este contexto que nos rodea, tampoco es de extrañar que el cibercrimen sea una tendencia completamente al alza. Según reflejan los datos publicados por el Ministerio del Interior en el "Estudio Sobre la Cibercriminalidad en España" en 2016 [2] (figura 1.1), este tipo de delitos no para de aumentar. No obstante, la pauta no se repite en los catalogados como hechos esclarecidos ni en las detenciones o personas investigadas por ello. A diario, son robados miles de datos confidenciales y muchas otras informaciones de interés, interceptadas tanto en sus servidores como durante los procesos de transmisión. Estos datos pueden ser lo que sea de quien sea. Desde conversaciones y archivos personales de todo tipo usados para la extorsión hasta los planes estratégicos de las empresas en el ámbito del espionaje industrial. Todo es información valiosa para un cracker.

Fig1.1 Evolución de los hechos cibercriminales en España
Fuente: Ministerio del Interior de España [2]

Es necesario decir que de ahora en adelante va a usarse la palabra craker para hacer referencia al atacante, a la persona que supuestamente intentaría penetrar en el sistema que se analiza. No va a usarse la palabra hacker para no darle una connotación negativa al término, dado que por muchos es una respetada profesión con fines alejados de la delincuencia.

Es por todo esto que la ciberseguridad ha ganado importancia hasta convertirse en un aspecto fundamental en la vida de cualquiera. No obstante, también es para la gran mayoría una inexorable llanura de desconocimiento, llena de palabras y conceptos de muy difícil compresión. Es una rareza encontrar a alguien que no trabaje en el sector informático que se haya cuestionado alguna vez preguntas como "¿Los mensajes de whatsapp se envían encriptados?" o "¿Que puertos hay abiertos en el router de mi hogar?", pero esto no debe ser excusa para eclipsar su relevancia. Tampoco es de extrañar, dado que para entender la ciberseguridad de verdad hay que inferir en conceptos tan diversos como las matemáticas avanzadas para encriptar o internet y sus embrollados protocolos de comunicaciones. Las empresas gastan descomunales cifras de dinero para realizar auditorías que garanticen que sus sistemas conectados a la red están "libres de vulnerabilidades" y que son "seguros". Esto es ir un paso más allá a tener un buen programa antivirus que periódicamente realice un barrido en busca de malware, puertas traseras y virus, se trata de realizar una autoría completa a los puertos a través de los cuales la máquina se comunica y intercambia información con la red. Tener estos accesos bien vigilados y, como bien se ha mencionado, "libres de vulnerabilidades", es primordial para garantizar cierto nivel de protección del equipo, dado que es allí adonde se encuentran las brechas que pueden aprovechar los virus y los crackers entre muchos otros con fines no beneficiosos para nosotros.

Se han añadido las comillas a los términos "seguridad" y "libre de vulnerabilidades" por qué hay que entender que en la práctica ningún sistema es 100% seguro. Con la cantidad adecuada de tiempo y recursos cualquier red del mundo es penetrable. Cuando se usa el término seguro, se quiere decir libre de vulnerabilidades conocidas, es decir, que el sistema no podrá ser atacado directamente desde el exterior usando métodos y caminos que, a pesar de ser conocidos y estudiados por muchos, suelen ser muy eficaces en multitud de situaciones. "Seguro" no significa que ningún cracker pueda acceder, sino simplemente que no va a serle fácil y que sus métodos necesitarán ser algo más ingeniosos.

Así es también como la ciberseguridad se ha convertido en un negocio con unos beneficios de crecimiento exponencial. Los profesionales de este campo, conscientes de lo valiosas y poco comunes que son sus habilidades, son contratados por las grandes compañías, dado que son las únicas que pueden permitirse sus servicios.

¿Y qué pasa con el usuario independiente? ¿Está condenada la pequeña red privada a estar desprotegida frente a crakers y boots? Por suerte, hoy en día hay herramientas gratuitas y muy completas que pueden ayudar al usuario con cierto nivel de conocimiento a guiarse por toda esta anarquía.

1.1 PROGRAMAS DE ESCANEO DE RED

- Open Vulnerability Assessment System, más conocido como OpenVAS, es un software desarrollado por la compañía Greenbone y soportado por una extensa comunidad de desarrolladores que lo actualizan periódicamente. Posee una buena interfaz de interacción con el usuario y es capaz de localizar una muy amplia gama de vulnerabilidades en los puertos de la red que este analizando. No obstante, esto no libra al usuario de necesitar un profundo conocimiento técnico sobre informática, redes y ciberseguridad en sí.

- Microsoft Baseline Security Analyzer, es la herramienta que Microsoft pone a disposición de sus clientes de manera gratuita para buscar vulnerabilidades en las pequeñas redes. Tiene una buena interfaz y prácticamente no requiere conocimientos técnicos para ejecutarlo y entender las conclusiones. El gran fallo de este software, aparte de que solo permite la búsqueda de vulnerabilidades en tu propia red, es que solo funciona con dispositivos Windows. Eso hace que el escaneo no funcione frente a dispositivos que usen otros sistemas operativos.

- Nmap, entre otros, es un programa de código abierto ejecutable desde la consola basado en el port scanning, muy usado hoy en día por auditores y analistas de ciberseguridad profesionales encargados de evaluar el nivel de protección de una red o de escanear una Ip. Des del punto de vista del craker, Nmap es una poderosa herramienta de inteligencia dada la enorme cantidad de información que es capaz de sustraer de forma muy sigilosa de cualquier Ip del mundo, estando en la misma red privada o no. Toda esta información es, sin duda alguna, muy útil para planificar y descubrir la manera de ejecutar el ataque que se crea conveniente. Aunque aparentemente usar la herramienta no es complicado, dado que solo es necesario conocer los comandos adecuados del programa para conectar con los puertos de la Ip de la víctima, sí que es de difícil interpretar sus resultados. Hacer un escaner en busca de vulnerabilidades con Nmap a una Ip, puede derivar fácilmente en un resultado del orden de 25 millones de líneas (considerando que en esa Ip solo haya entre 2 y 4 puertos abiertos, este número podría ser mucho más elevado). A través de los ojos de un usuario con conocimiento medio (englobando todos los usuarios de la red del mundo, no solo los de Nmap) intentar saber, por ejemplo, si su red privada es segura usando Nmap parece una tarea imposible. Además, hacen falta conocimientos técnicos bastante complejos sobre el programa y la habilidad de

guiarse por la consola, cosa que no es común.

Como bien puede observarse, no hay muchas alternativas para hacer llegar la ciberseguridad a las pequeñas redes propiedad de usuarios despreocupados del tema, dado que en general son muy limitadas o requieren de conocimientos técnicos que implican horas de dedicación. Todo esto ha derivado en la expansión de la despreocupación sobre la ciberseguridad entre los usuarios medios de la red, dado que es una postura más cómoda y fácil de mantener que la de invertir tiempo o recursos monetarios.

1.2 LA MENTALIDAD DEL USUARIO MEDIO Y LOS ANTIVIRUS

El usuario medio, entendiéndolo como el usuario más habitual de la red (persona que por ejemplo usa su smartphone para frecuentar las redes sociales y que raramente usa internet para otros fines), suele tener instalado en su ordenador algún tipo de programa antivirus. La gama de softwares gratuitos que ofrecen un servicio de bloqueo de descarga de archivos cuando estos llevan algún tipo de malware asociado y que escanean la maquina en busca de virus entre otras cosas de forma eficiente y automatizada es muy amplía. Además, sus resultados suelen ser lo prometido. No obstante, estos programas no garantizan la seguridad, ya que sus objetivos son los de detectar virus y amenazas ya presentes en el equipo o evitar la entrada directa mediante descarga de estas. En ningún momento se dedican a localizar las vulnerabilidades que el craker o el virus pueden aprovechar para penetrar en el sistema y ejecutar acciones posteriores frente a las cuales el dispositivo va a estar completamente desprotegido. Estas amenazas pasan inadvertidas a los antivirus porque no son ningún tipo de código malicioso que pueda ser leído y interpretado como algo nocivo. Son brechas en la seguridad, caminos que pueda llevar a que desde el exterior algo o alguien obligue a la maquina a ejecutar acciones generalmente perjudiciales para ella o para un tercero.

1.3 DEFINICIÓN DE VULNERABILIDAD

Según Daniel Maldonado, autor de “Diccionario de hacking” [3] entre muchos otros libros y artículos de ciberseguridad, una vulnerabilidad se define de la siguiente manera:

“Es una debilidad en algún software, hardware o procedimiento que puede permitir a un atacante realizar acciones que, normalmente, no tiene permitidas. En general, tiene como raíz la ausencia o la debilidad en uno o más controles.”

Cuando estos softwares, hardwares o procedimientos están conectados a través de un puerto de red a internet, es cuando estas vulnerabilidades se convierten en potenciales amenazas, ya que es cuando los atacantes tienen acceso a ellas mediante el mencionado puerto de red. Es este punto el principal problema de los antivirus, que no buscan en los puertos red la existencia de estas puertas abiertas a los crakers. El tipo de programa que si lo haría son los softwares de escaneo de redes definidos anteriormente. No obstante, ya se han hecho notar sus fortalezas y flaquezas derivadas de la falta de cultura en ciberseguridad.

2. OBJETIVOS Y PROPUESTA DE SOLUCIÓN

A continuación, van a detallarse los objetivos del proyecto y la propuesta de solución a la problemática anteriormente expuesta:

2.1 OBJETIVOS

Con el objetivo solventar la falta de seguridad en las redes privadas dada la dificultad que entraña realizar un examen en busca de vulnerabilidades en los puertos red, se va a realizar un aplicativo. Dicho aplicativo debe automatizar un escaneo completo en la Ip de la víctima en busca de vulnerabilidades usando Nmap como herramienta, dejando a disposición del usuario un documento PDF resumiendo las conclusiones del escaneo usando un lenguaje que pueda llegar a buena parte de la población, es decir, que no requiera de conocimientos técnicos.

2.2 PROPUESTA DE SOLUCIÓN

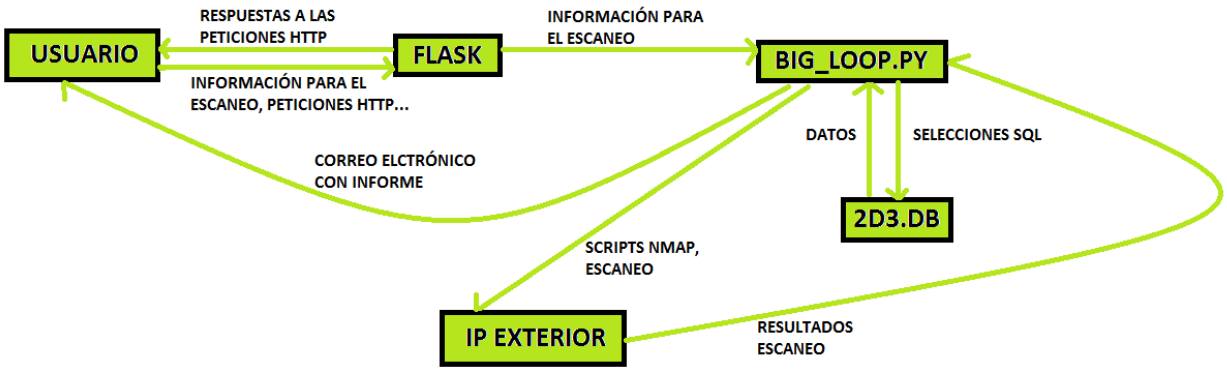


Fig 2.2.1 Esquema interacción componentes VULNInterpreter

En la imagen puede observarse un esquema de VULNInterpreter (figura 2.2.1), un sistema que mantiene una aplicación que, como bien se ha mencionado tiene como objetivo el envío de un informe resumiendo de forma comprensible las posibles vulnerabilidades existentes en los puertos red de la Ip específica. La aplicación está hecha con Python aunque se usan otros lenguajes de programación. Hay que remarcar también el uso que se hace de Nmap como herramienta de port scaninng, elegida entre varias opciones por las razones que se presentan a continuación.

2.3. EXPLICACIÓN ESQUEMA VULNInterpreter

La figura muestra la interacción completa entre los distintos entes que componen el sistema. El usuario interactúa con el servidor flask (una popular librería de Python para hacer servidores) a través de peticiones y respuestas http. Este, si se dan las condiciones adecuadas, facilita los datos necesarios y ejecuta big_loop.py, un código Python que se combina con Nmap y con una base de datos SQL (2D3.DB) para ejecutar el escaneo en la Ip objetivo especificada, filtrar los resultados de ese escaneo, generar el informe con las vulnerabilidades encontradas y devolvérselo al usuario.

2.4. PORQUE NMAP

Se elijó Nmap como método para realizar el port scanning por qué, a pesar de ser una herramienta compleja cuyo correcto entendimiento ha requerido horas, es también la favorita de muchos hackers y analistas profesionales, por su sigilo y eficacia realizando estos análisis. El popular hacker español Chema Alonso, considerado uno de los mejores hackers éticos por la DEFCON, la convención hacker más grande del mundo, ha escrito diversos artículos sobre Nmap en su blog, www.elladodelmal.com. Análogamente, los analistas de la reputada empresa en ciberseguridad ESET, hablan de ella y de cómo la usan en las auditorias de ciberseguridad que hacen en el blog www.welivesecurity.com. Además, el hecho de que sea ejecutable desde consola lo hace idóneo para recibir, filtrar y interpretar sus resultados (a pesar de que puedan alargarse en una pequeña red privada hasta los 30 millones de líneas o más) con el código Python y las demás herramientas usadas.

2.5 ALCANCE Y LIMITACIONES DEL PROYECTO

El proyecto va a centrarse en la creación de un software que localice vulnerabilidades. No se pretende dar una solución a ellas, sino hacer notar al usuario su presencia. La gama de posibles vulnerabilidades es tan extensa como variados son los métodos usados para sacarles provecho. Es por eso que dar una solución para cada fallo de seguridad que se encuentra no es una tarea factible, dado que esta dependerá en enorme medida de circunstancias de cada equipo, aplicación que causa la vulnerabilidad, configuraciones propias del usuario, etc. En lugar de eso, la intención es dejar clara constancia de que la vulnerabilidad se encuentra allí, mostrando también una breve descripción ella que contenga las palabras clave que puedan permitir al usuario hacerse una idea de la situación para que pueda iniciar una búsqueda de posibles soluciones para su caso concreto.

Además, en caso de encontrarse vulnerabilidad, también se proporcionará el link directo a la página web de Nmap al apartado del script en concreto que haya localizado esa vulnerabilidad.

3. DESARROLLO Y FUNCIONAMINETO DEL APLICATIVO

A continuación, se expondrá un resumen de los distintos componentes y fases de desarrollo del programa, siguiendo el orden cronológico en el cual fueron desarrollados o estudiados.

3.1. NMAP, LA RED Y EL ESCANEO DE PUERTOS

3.1.1. NMAP

Nmap fue desarrollado por Gordon Lyon, un conocido y respetado experto en seguridad de redes que responde al pseudónimo hacker de Fyodor Vaskovich. Está escrito en varios lenguajes entre los cuales destacan Python, C++ y Lua. Actualmente, se encuentra en estado de constante desarrollo por una extensa comunidad que trabaja a diario para incrementar el número de vulnerabilidades que detecta y ampliar su gama de utilidades.

Lo que Nmap hace es un escaneado de puertos en la Ip objetivo, es decir, le manda mensajes a través de distintos protocolos de comunicación (http, smtp, ssh...), solicitando distintas respuestas (REQUEST, USER, ...), y analizando parámetros concretos dependiendo de lo que este rastreando. Con toda esta información que extrae, después de ser filtrada con el script de extensión NSE (Nmap Script Engine) adecuado, el software es capaz haberse formado una idea muy realista de lo que hay o no hay en esa Ip.

Para sacarle el máximo partido, Nmap organiza sus scripts por categorías en un motor de búsqueda llamado NSearch, que facilita muchísimo que el usuario pueda encontrar el script adecuado para la tarea concreta que precise. Estas categorías son:

- **Auth:** busca la existencia de usuarios y claves de los distintos servicios que puedan estar ejecutándose en los puertos.
- **Discovery:** busca la existencia de servicios en ejecución en los puertos (Servidor MySQL, Servidor Apache…).
- **Malware:** Busca la presencia de malware y puertas traseras en el sistema.

- **Intrusive:** ejecuta los scripts que más penetran en la víctima. Hay que tener presente que una mayor penetración incluye en la mayoría de los casos un aumento del rastro que se deja por parte del atacante.

- **Default:** ejecuta los scripts más básicos que tiene Nmap.

- **All:** en este caso se ejecutarían todos los scripts que NSE posee.

- **Vuln:** ejecuta los scripts centrados en detectar la presencia de vulnerabilidades.

Dada la naturaleza del proyecto, la categoría que va a usarse será Vuln, para obtener directamente toda la información relativa a las vulnerabilidades de la Ip objetivo. El escaneo de puertos de Nmap puede realizarse mediante los protocolos TCP y UDP, los cuales vamos a detallar a continuación.

3.1.2. LA RED Y LOS PROTOCOLOS TCP Y UDP

Los puertos, que como bien se ha mencionado anteriormente son la forma en como los sistemas aislados se conectan con la red, pueden hacerlo mediante dos protocolos, TCP y UDP:

TCP

El protocolo de control de transmisión o “Transmission Control Protocol” es el protocolo de comunicación más usado. Este actúa de capa intermedia entre internet y la aplicación, garantizando que la información llegará a su destino igual que se envió, en el mismo orden y sin ningún error. Las aplicaciones que usan HTTP, como la mayoría de las páginas web, o SMTP, como la mayoría de los servicios de correo electrónico, utilizan la conexión TCP para el intercambio de datos.

Para establecer estas conexiones (figura 3.2.1.1), el cliente envía un paquete de datos SYN hasta el servidor. Este intentará validar la conexión enviando al cliente un paquete SYN/ACK. El cliente deberá responder con un paquete ACK para que la conexión quede establecida. Esta secuencia de validación SYN/ACK debe llevarse a cabo en todo momento de la comunicación para asegurar que el cliente recibe todos los datos en el mismo orden que se envían. Esto significa que cada paquete que va del servidor al cliente deberá ser validado por parte de este último con un paquete ACK. Si no es así, el servidor seguirá enviando el mismo paquete hasta obtener la confirmación de que el cliente lo ha recibido. Por todo esto se dice que un protocolo orientado a la conexión.

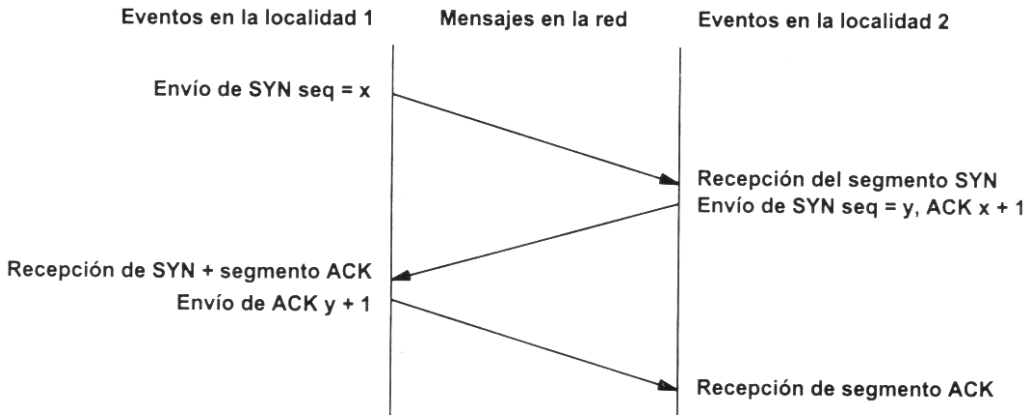


Fig 3.2.1.1 Establecimiento de una conexión TCP
Fuente: Universidad Autónoma del Estado de Hidalgo [4]

UDP

El Protocolo de Datagramas de Usuario o "User Datagram Protocol" envía los paquetes de datos con la información sin validar la conexión entre cliente y servidor ni controlarlos una vez enviados, a diferencia del protocolo TCP que primero debe validar la conexión y no mandará ningún paquete nuevo hasta tener la confirmación de que se ha recibido el anterior. En el protocolo UDP los paquetes ya contienen la información suficiente para que estos lleguen a su destino. Por esto se considera un protocolo orientado al mensaje. Aunque aparentemente pueda presentar desventajas respecto TCP, estas lo hacen más eficiente, por lo tanto, un servidor gestionará mejor un volumen elevado de peticiones UDP que de TCP. Esto lo hace idóneo para todos aquellos procesos en los que la velocidad es más relevante que la seguridad de que no se haya perdido o desordenando ningún dato, como en transmisiones de audio o vídeo. Aplicaciones como el DNS (Domain Name System) son soportadas por UDP.

Dado el gran uso de los puertos de estos dos protocolos (aunque de TCP en mucha mayor medida), es necesario hacer el escaneo de puertos utilizando ambos métodos de comunicación para localizar todas las vulnerabilidades.

3.1.3. REALIZACIÓN DEL ESCANEO

Después de realizar un profundo análisis de Nmap y de sus posibles escaneos y configuraciones de estos sobre una IP, se llega a la conclusión que en un examen

muy completo sobre las vulnerabilidades presentes en los puertos (tanto los del protocolo TCP como los del protocolo UDP) del objetivo serán necesarios los siguientes parámetros:

ESPECIFICACIONES DEL ESCANEO

- -sS: este parámetro indica que el escaneo de puertos debe realizarse mediante la técnica TCP SYN. En este tipo de escaneados se intenta localizar los puertos accesibles enviándoles peticiones SYN. En caso de recibir una respuesta con un paquete SYN/ACK, significa que el puerto está abierto. En tal caso, se responde con un paquete RST, para abortar la conexión, en vez de con un paquete ACK, que la validaría y muy probablemente dejaría rastro de nuestra presencia en el objetivo. La otra técnica válida para llevar a cabo el tipo de escaneo que se pretende es TCP conncet. Al inicio este método y el anterior son iguales, no empiezan a diferir hasta el final, dado que mediante TCP conncet se respondería al paquete SYN/ACK mandado por el servidor con un paquete ACK, hecho que dejaría establecida la connexión TCP. Esto dejaría muy probablemente pruebas registradas de las acciones llevadas a cabo durante el escaneo, cosa que no interesa des del punto de vista de un atacante. Por lo tanto, a pesar de que TCP connect es el método más rápido y simple, va a usarse TCP SYN.

- -sU: este parámetro indica que el escaneo de puertos debe realizarse mediante UDP. En este caso, el objetivo serán los puertos que usen este protocolo y los intentará localizar comunicándose con ellos a través de él.

- -vv: este parámetro ordena a Nmap a mostrarnos toda la información relativa a los puertos que ha encontrado abiertos.

- -d9: este parámetro eleva al máximo el nivel de depuración, el 9 en este caso. Con esto indicamos a Nmap que debe pasar por todos y cada uno de los puertos de la Ip para asegurar que ningún puerto abierto al que se pueda acceder queda escondido.

- -f: este parámetro indica a Nmap que debe enviar los paquetes fragmentados. La fragmentación de paquetes IP es un procedimiento muy habitual cuando estos sobrepasan un cierto tamaño. Simplemente los paquetes se dividen y se reconstruyen al llegar. En este caso, a pesar de que no es necesario por tamaño, puede ayudar a saltarse el firewall que protege los puertos y que podría esconder puertos abiertos no accesibles a

simple vista por que están destinados a aplicativos de la red privada. Se añade este comando para maximizar el número de puertos abiertos localizados, que es dónde residen las vulnerabilidades.

- --script vuln: este comando ejecuta los scripts de la categoría vuln adecuados en función del tipo de aplicación detectada en el puerto abierto. Cada script está especializado en la búsqueda de una vulnerabilidad concreta, las cuales se detallan a continuación en una tabla elaborada por el autor del trabajo en la cual intenta explicar de forma resumida y en un lenguaje coloquial cada una de las vulnerabilidades que el software puede localizar. Dichas definiciones serán usadas para crear la base de datos de la cual se extraerá información para redactar los informes. Por lo tanto, con ellas se busca dar al usuario una idea general de la situación y las palabras clave para que este pueda iniciar una búsqueda en profundidad sobre el tema.

afp-path-vuln	Intento de ataque de directorio transversal. El atacante podría acceder a los directorios superiores del servidor y visualizar sus contenidos dada una mala validación de los usuarios que pueden acceder. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2010-0533.
broadcast-avahi-dos	Vulnerabilidad DoS presente en Avahi (Denegación de Servicio). Si la configuración del multicast (grupos de equipos que reciben la misma comunicación simultáneamente) es Avahi (un popular software usado para facilitar las comunicaciones), se puede usar el servicio DNS (Domain Name System, relaciona los nombres de los servicios web con las IP dónde residen) para localizar usuarios a los cuales ejecutar un ataque de denegación de servicio o DoS, en el cual la víctima podría quedar fuera de la red o incluso inoperativa. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2011-1002.
clamav-exec	Ejecución remota de comandos en el servidor ClamAV. ClamAV es un popular software antivirus

	que en muchas de sus versiones permite la ejecución de comandos sin autenticar el usuario.
distcc-cve2004-2687	Posible tipo de ejecución remota de código en la comunicación. Reportado como vulnerabilidad común en la NVD con el nombre CVE2004-2687.
dns-update	Ejecución de una asignación de IP a un dominio por parte de un usuario sin autenticar. Esta operación se haría usando el servicio DNS Dinámico, que actualiza la información relativa a la asignación de IP y nombres en el servidor.
firewall-bypass	Intento de saltarse el cortafuegos.
ftp-libopie	Búsqueda de las vulnerabilidades comunes reportadas en la NVD con el nombre CVE-2010-1938.
ftp-proftpd-backdoor	Búsqueda de la puerta trasera ProFTPD 1.3.3c (un tipo bastante común de puerta trasera).
ftp-vsftpd-backdoor	Búsqueda de la presencia de la puerta trasera vsFTPd 2.3.4 (un tipo bastante común de puerta trasera). Reportado como vulnerabilidad común en la NVD con el nombre CVE-2011-2523.
ftp-vuln-cve2010-4221	Ataque al bufer de datos del servidor ProFTPD. Usando la secuencia de comunicación TELNET_IAC, del proceso proftpd, se puede dejar expuesto el buffer (una memoria temporal que almacena datos durante las comunicaciones), lo que da al atacante la posibilidad de ejecutar código de su elección. Además, no se requiere ningún tipo de identificación para explotar esta vulnerabilidad. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2010-4221.

http-adobe-coldfusion-apsa1301	Posible bypass en la autenticación. El atacante puede acceder a la aplicación o web sin el usuario ni la contraseña.
http-aspnet-debug	Localizador del proceso de debugging. El proceso de debugging o depuración consiste en la localización de errores y cosas fuera de lo normal en la aplicación o web, lo que podría dar al atacante información muy valiosa.
http-avaya-ipoffice-users	Localizador de usuarios de Avaya IP Office systems. Avaya es un software especializado de comunicaciones. El atacante podría acceder con estos nombres de usuarios al servidor, entre otras posibilidades.
http-awstatstotals-exec	Vulnerabilidad en el archivo AWStats Totals. El archivo AWStats Totals es un archivo PHP donde se pueden ver los visitantes, el número de visitas entre mucha otra información aprovechable por el atacante. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2008-3922.
http-axis2-dir-traversal	Vulnerabilidad de directorio transversal en Apache Axis2. Enviando la petición adecuada, en algunas versiones de este servidor se puede recibir el usuario y la contraseña del administrador en la respuesta
http-cookie-flags	Examen de las cookies del servicio HTTP. Búsqueda de cookies sin httponly flag (en tal caso, significa que es fácil acceder a las cookies protegidas) y de cookies ssl que no tengan activada la secure flag (en tal caso, pueden ser fácilmente interceptadas y leídas).
http-cross-domain-policy	Consulta del archivo crossdomain.xml y clientaccesspolicy.xml. Se buscan listas de dominios

	conocidos y seguros. Esta información puede usarse para múltiples usos, como buscar objetivos para ataques CSRF. El CSRF o “Cross Site Request Forgery”, es un ataque que consiste en forzar al navegador de la víctima a hacer peticiones en sitios web donde se haya dado de alta para captar las credenciales de acceso entre otra información valiosa.
http-csrf	Búsqueda de vulnerabilidades CSRF. El CSRF o “Cross Site Request Forgery”, es un ataque que consiste en forzar al navegador de la víctima a hacer peticiones en sitios web donde se haya dado de alta para captar las credenciales de acceso entre otra información valiosa.
http-dlink-backdoor	Búsqueda de puerta trasera en routers D-Link. Actuando sobre el firmware (software que gestiona la lógica d control de los circuitos electrónicos) se puede hacer un bypass en la autenticación de usuarios y acceder al router.
http-dombased-xss	Búsqueda de vulnerabilidades que afecten a la ejecución del código JavaScript.
http-enum	Búsqueda de directorios populares.
http-fileupload-exploiter	Búsqueda de vulnerabilidades en las opciones de descargas. Se analizan los métodos de descarga en distintas condiciones para buscar puntos que puedan presentar inseguridades aprovechables por el atacante.
http-frontpage-login	Búsqueda de objetivos FrontPage. FrontPage es un software de Microsoft usando para crear páginas web y aplicaciones. En las condiciones adecuadas el atacante puede modificarlas y acceder al servidor.

http-git	Búsqueda en el repositorio Git (aplicativo de control de versiones de un sofware) de aplicaciones con información comprometedora.
http-huawei-hg5xx-vuln	Búsqueda de modems Huawei HG530x, HG520x, HG510x. Busca vulnerabilidades en estos módems como el posible acceso sin credenciales y sus posibles fallos. Además, extrae cantidad de información relativa a los protocolos PPPoE (Point-to-Point over Ethernet), que sirven para conectar múltiples usuarios de una red local a un lugar remoto.
http-iis-webdav-vuln	Búsqueda de vulnerabilidades en IIS 5.1/6.0 (Internet information Services, aplicativo Microsoft usado para convertir un PC en un servidor). En caso de encontrarla, el atacante podría acceder a carpetas restringidas del servidor.
http-internal-ip-disclosure	Intento de descubrir la Ip interna del servidor enviando una petición HTTP/1.0.
http-jsonp-detection	Búsqueda de vulnerabilidades en JSONP. JSONP es una técnica de comunicación de los programas JavaScript. En caso de resultar vulnerable, el atacante podría hacer colar sus archivos en el servidor víctima haciéndole creer que son originarios de su propio sistema.
http-litespeed-sourcecode-download	Búsqueda de la vulnerabilidad null-byte poisoning en servidores Litespeed. En caso de darse, el atacante podría acabar teniendo acceso completo a los archivos del sistema. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2010-2333.
http-majordomo2-dir-traversal	Búsqueda de la vulnerabilidad de directorio transversal en Majordomo2, un popular software de correo electrónico. Reportado como vulnerabilidad

	común en la NVD con el nombre CVE-2011-0049.
http-method-tamper	Búsqueda de vulnerabilidades en los métodos del protocolo HTTP.
http-passwd	Intento de ataque de directorio transversal accediendo a los archivos de la BIOS (el paso previo al sistema operativo en la arquitectura del computador) o a los archivos que contienen la información de los usuarios del sistema operativo. En un ataque de directorio transversal, el atacante puede acceder a los directorios superiores del servidor y visualizar sus contenidos.
http-phpmyadmin-dir-traversal	Intento de ataque de directorio transversal al software phpMyAdmin. En un ataque de directorio transversal, el atacante puede acceder a los directorios superiores del servidor y visualizar sus contenidos.
http-phpself-xss	Búsqueda de archivos PHP vulnerables a un ataque Cross-Site-Scripting. En este tipo de ataques, el atacante puede inyectar en los archivos vulnerables líneas de código, modificando los scripts de las aplicaciones o páginas web.
http-shellshock	Verificación presencia vulnerabilidad shellshock. Esta vulnerabilidad presente en el intérprete Bash, el intérprete de comandos más popular para sistemas operativos con base UNIX (Android, Linux, Mac OS X…), permitiría al atacante ejecutar comandos a su voluntad. Reportado como vulnerabilidad común en la NVD con los nombres CVE-2014-6271 y CVE-2014-7169.
http-slowloris-check	Verificación de vulnerabilidad frente a ataques DoS. En caso de serlo, el atacante puede ejecutar un ataque DoS (“Denial of Service”) o de denegación de servicio, que consiste en inundar al objetivo de

	peticiones para saturar su capacidad de procesamiento, dejando a los dispositivos fuera de la red o incluso completamente inoperables según la naturaleza del ataque.
http-sql-injection	Búsqueda de comunicaciones HTTP vulnerables a un ataque SQL Injection. En este tipo de ataques el atacante puede inyectar instrucciones SQL (popular lenguaje de programación para crear y administrar bases de datos) a su voluntad.
http-stored-xss	Búsqueda de objetivos para ataque Cross-Site-Scripting. En este tipo de ataques, el atacante puede inyectar en los archivos vulnerables líneas de código, modificando los scripts de las aplicaciones o páginas web.
http-tplink-dir-traversal	Explotación de la vulnerabilidad de directorio transversal en routers inalámbricos TP-Link. En estos ataques, el atacante puede acceder a los directorios superiores del servidor y visualizar sus contenidos dada una mala validación de los usuarios que pueden acceder
http-trace	Búsqueda de si el método HTTP TRACE esta activado. En caso de estarlo, el atacante podría visualizador los cambios realizados en las peticiones HTTP por parte del servidor, es decir, se interceptaría parte de la petición o repuesta HTTP.
http-vmware-path-vuln	Búsqueda de la vulnerabilidad de directorio transversal en tecnología VMWare ESX, usada para crear medios de computación virtuales. En caso de darse la vulnerabilidad, significa que hay una mala validación de los usuarios que pueden acceder a los directorios superiores de las aplicaciones, dejando vía libre de acceso al atacante. Reportado como vulnerabilidad común en la NVD con el nombre CVE-

	2009-3733.
http-vuln-cve2006-3392	Búsqueda de las vulnerabilidades comunes reportadas en la NVD con el nombre CVE-2006-3392.
http-vuln-cve2009-3960	Búsqueda de las vulnerabilidades comunes reportadas en la NVD con el nombre CVE-2009-3960, también conocidas como Adobe XML External Entity Injection. Adobe XML External Entity Injection hace referencia a una amplia variedad de vulnerabilidades basadas en el uso del lenguaje XML.
http-vuln-cve2010-0738	Intento de bypass en la autenticación de usuarios en servidores que usen software JBoss. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2010-0738.
http-vuln-cve2010-2861	Búsqueda de la vulnerabilidad de directorio transversal en aplicaciones basadas en el software ColdFusion. En caso de ser vulnerable el atacante podría acceder a los directorios superiores del servidor y visualizar sus contenidos dada una mala validación de los usuarios que pueden acceder. Reportado como vulnerabilidad común en la NVD con el nombre CVE2010-2861.
http-vuln-cve2011-3192	Búsqueda de la posibilidad de que un servidor Apache, uno de los tipos de servidor más usados del mundo, sea vulnerable a un ataque DoS. En caso de serlo, el atacante puede ejecutar un ataque DoS (“Denial of Service”) o de denegación de servicio, que inunda el objetivo de peticiones para saturar su capacidad de procesamiento, dejando a los dispositivos fuera de la red o incluso completamente inoperables según la naturaleza del ataque. Reportado como vulnerabilidad común en la NVD

	con el nombre CVE2011-3192.
http-vuln-cve2011-3368	Intento de bypass en la autenticación del servidor Reverse Proxy. Este servidor suele estar situado detrás del firewall de la red privada y relaciona peticiones de los clientes con los correspondientes servicios disponibles en la red privada. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2011-3368.
http-vuln-cve2012-1823	Búsqueda de vulnerabilidades en instalaciones PHP-CGI. El CGI ("Common Gateway Interface") es un servicio para montar páginas web y aplicaciones dinámicas. En caso de darse, el atacante podría llegar a ejecutar código de forma remota. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2012-1823.
http-vuln-cve2013-0156	Búsqueda de vulnerabilidades en servidores que usen el popular software para gestión y desarrollo de aplicaciones Ruby on Rails. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2013-0156.
http-vuln-cve2013-6786	Búsqueda de la vulnerabilidad frente a ataques Cross-Site-Scripting en aplicaciones basadas en el servidor Allegro RomPager. En este tipo de ataques, el atacante puede inyectar en los archivos vulnerables líneas de código, modificando los scripts de las aplicaciones o páginas web. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2013-6786.
http-vuln-cve2013-7091	Búsqueda de las vulnerabilidades comunes reportadas en la NVD con el nombre CVE-2013-7091.
http-vuln-cve2014-2126	Búsqueda de la posibilidad de que Cisco ASA

	(popular hardware para montar infraestructura de red) sea vulnerable a un escalonamiento de privilegios en el servicio de monitoreo ASDM. ASDM es un popular servicio de configuración y monitoreo de dispositivos ASA. En caso de tener esta vulnerabilidad, el atacante podría acceder a las mismas herramientas y pantallas de ASDM que un usuario con todos los privilegios. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-2126.
http-vuln-cve2014-2127	Búsqueda de la posibilidad de que Cisco ASA (popular hardware para montar infraestructura de red) sea vulnerable a un escalonamiento de privilegios en el servicio de comunicación SSL de VPN. En caso de serlo, el atacante solo necesitaría ser un usuario válido, pero sin privilegios para ejecutar acciones privilegias sobre las comunicaciones SSL (protocolo de encriptación) de las redes virtuales privadas o VPN, normalmente usadas para proteger zonas sensibles de la red local o que necesitan ser accedidas desde el exterior de forma segura. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-2127.
http-vuln-cve2014-2128	Búsqueda de la posibilidad de que Cisco ASA (popular hardware para montar infraestructura de red) sea vulnerable a un by-pass en la autenticación de los servicios SSL de VPN. En caso de serlo, el atacante podría acceder a las comunicaciones SSL (protocolo de encriptación) de las redes virtuales privadas o VPN, normalmente usadas para proteger zonas sensibles de la red local o que necesitan ser accedidas desde el exterior de forma segura. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-2128.
http-vuln-cve2014-2129	Búsqueda de la posibilidad de que Cisco ASA (popular hardware para montar infraestructura de

	red) sea vulnerable a un ataque DoS. En caso de serlo, el atacante puede ejecutar un ataque DoS ("Denial of Service") o de denegación de servicio, que inunda el objetivo de peticiones para saturar su capacidad de procesamiento, dejando a los dispositivos fuera de la red o incluso completamente inoperables según la naturaleza del ataque. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-2129.
http-vuln-cve2014-3704	Búsqueda de la vulnerabilidad conocida 'Drupageddon'. Drupal es uno de los sistemas de gestión para páginas web dinámicas más populares del mundo. En caso de darse esta vulnerabilidad, el atacante podría llevar acabo un ataque de sql-injection con el objetivo de extraer información directamente de las bases de datos. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-3704.
http-vuln-cve2014-8877	Búsqueda de vulnerabilidades relacionadas con la ejecución remota de código en Wordpress. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-8877.
http-vuln-cve2015-1427	Búsqueda de las vulnerabilidades reportadas en la NVD con el nombre CVE-2015-1427. En caso de darse, el atacante podría ganar facilidad para ejecutar código de forma remota.
http-vuln-cve2015-1635	Búsqueda de la vulnerabilidad MS15-034 en sistemas Microsoft Windows. En caso de darse, el atacante podría tener libertad para ejecutar código a distancia. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2015-1635.
http-vuln-cve2017-1001000	Búsqueda de vulnerabilidades Wordpress 4.7.0 y 4.7.1. En caso de darse, el atacante podría tener libertad para inyectar contenido en los posts.

	Reportado como vulnerabilidad común en la NVD con el nombre CVE-2017-1001000.
http-vuln-cve2017-5638	Búsqueda de la vulnerabilidad Apache Struts Remote Code Execution Vulnerability. En caso de darse, el atacante podría tener facilidad para ejecutar código de forma remota en el servidor. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2017-5638.
http-vuln-cve2017-5689	Búsqueda en dispositivos de Intel Active Management Technology de la vulnerabilidad INTEL-SA-00075 privilege escalation. Resportado como vulnerabilidad común en la NVD con el nombre CVE-2017-5689.
http-vuln-cve2017-8917	Búsqueda de las vulnerabilidades comunes reportadas en la NVD con el nombre CVE-2017-8917, varias de ellas relacionadas con ataques SQL injection.
http-vuln-misfortune-cookie	Búsqueda de la vulnerabilidad Misfortune Cookie en servidores que usen RoomPger como motor de protocolo HTTP. Esta vulnerabilidad podría llegar a permitir al atacante a tomar el control absoluto del router.
http-vuln-wnr1000-creds	Búsqueda de vulnerabilidades en routers de la serie WNR 1000 que podrían permitir al atacante tener los privilegios de administrador sobre este.
http-wordpress-users	Búsqueda de usuarios en un blog Wordpress.
ipmi-cipher-zero	Búsqueda de hardware susceptible a un acceso mediante bypass en el proceso de autenticación de usuario.
irc-botnet-channels	Búsqueda de bots maliciosos en los canales

	designados para la comunicación IRC, un protocolo que permite conversaciones grupales en tiempo real entre varios usuarios.
irc-unrealircd-backdoor	Búsqueda de puertas traseras en servidores de comunicación IRC. Estos servidores usan el protocolo de comunicación IRC que permite conversaciones grupales en tiempo real entre varios usuarios.
mysql-vuln-cve2012-2122	Intento de bypass en la autenticación de usuarios en servidores MySQL que usen tecnología MariaDB. Reportado como vulnerabilidad común en la NVD con el nombre CVE2012-2122.
netbus-auth-bypass	Intento de bypass en la autenticación de usuarios de los servidores NetBus, un software usado para usar por control remoto sistemas que funcionen con Windows.
puppet-naivesigning	Búsqueda de vulnerabilidades en servidores que usen el software de gestión Puppet. Puppet es una herramienta que permite la fácil administración de la configuración de sistemas operativos que usan UNIX como base.
rdp-vuln-ms12-020	Búsqueda de vulnerabilidades en servicios de escritorio remoto de Microsoft. Reportadas en la NVD con los nombres CVE-2012-0152 y CVE-2012-0002.
realvnc-auth-bypass	Intento de bypass en la autenticación de usuarios de servidores VNC. VNC es un popular software libre usado para poder ver acciones de un ordenador (servidor) a través de otro (cliente). Reportado como vulnerabilidad común en la NVD con el nombre CVE-2006-2369.

rmi-vuln-classloader	Intento de ejecución remota de código JavaScript. En caso de darse, el atacante podría llegar a tener total libertad para cargar y ejecutar en la máquina de la víctima código escrito en este lenguaje.
rsa-vuln-roca	Intento de recuperar la clave privada del protocolo de cifrado de las comunicaciones a través de la clave pública. En caso de darse, el atacante podría leer comunicaciones entre el usuario y la red sin ningún tipo de cifrado.
samba-vuln-cve-2012-1182	Intento de ejecución remota de código a través de Samba. Samba es un popular software usado para compartir archivos de forma fácil entre máquinas diferentes. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2012-1182.
smb-double-pulsar-backdoor	Búsqueda de la puerta trasera Double Pulsar. Esta puerta trasera fue diseñada y implantada por la NSA (National Security Agency). Es conocida por el público gracias a unas filtraciones producidas en 2017. En caso de encontrarse, lo que es más común de lo que uno podría esperar, el atacante la podría aprovechar para obtener un gran control sobre la máquina de una forma muy discreta.
smb-vuln-conficker	Búsqueda del gusano Conficker, un tipo de gusano bastante común y letal para la máquina que usa una brecha presente en algunas configuraciones de Windows para replicarse.
smb-vuln-cve-2017-7494	Búsqueda de las vulnerabilidades reportadas en la NVD con el nombre CVE-2017-7494, presentes en versiones de Samba, un popular software para compartir archivos entre máquinas diferentes.
smb-vuln-cve2009-3103	Búsqueda de objetivos Microsoft Windows vulnerables a un ataque DoS letal. En caso de serlo,

	el atacante puede ejecutar un ataque DoS (“Denial of Service”) o de denegación de servicio, que inunda el objetivo de peticiones para saturar su capacidad de procesamiento, dejando a los dispositivos fuera de la red o incluso completamente inoperables según la naturaleza del ataque. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2009-3103.
smb-vuln-ms06-025	Búsqueda de servidores RAS (“Remote Aces Server”, servidor de acceso remoto) vulnerables a un ataque MS06-025.
smb-vuln-ms07-029	Búsqueda de objetivos (servidores DNS en este caso) a los que practicar un ataque MS07-029. Este Ataque pretende corromper la configuración del servidor y realizar una extracción de información.
smb-vuln-ms08-067	Búsqueda de objetivos con sistemas Microsoft Windows vulnerables a MS08-067. En caso de darse la vulnerabilidad el atacante contaría con la posibilidad de ejecutar código en la Ip de la víctima de forma remota.
smb-vuln-ms10-054	Búsqueda de objetivos para ms10-054. El ms10-054 consiste en la corrupción de la memoria de la víctima a distancia por parte del atacante.
smb-vuln-ms10-061	Búsqueda de objetivos para ejecutar ms10-061. El ms10-061 consiste en suplantar la cola de impresión por completo, a la libre decisión del atacante.
smb-vuln-ms17-010	Búsqueda de vulnerabilidades en Microsoft SMBv1 (protocolo usado en sistemas operativos de Microsoft para compartir información entre nodos de una red). En caso de darse el atacante dispondría de una gran facilidad para ejecutar código a distancia.

smb-vuln-regsvc-dos	Búsqueda de la posibilidad de dejar el sistema inoperativo por una vulnerabilidad en el archivo regsvc de Microsoft Windows, un archivo que contiene bibliotecas y funciones necesarios para las páginas web y aplicaciones dinámicas.
smb-vuln-webexec	Búsqueda de la posibilidad de ejecución remota de código en servicios de video basados en el popular WebExService. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2018-15442.
smb2-vuln-uptime	Búsqueda de vulnerabilidades durante la negociación del cambio de llaves para el cifrado al inicio de una comunicación usando el protocolo SMB2 (protocolo usado en sistemas operativos de Microsoft para compartir información entre nodos de una red).
smtp-vuln-cve2010-4344	Búsqueda y explotación de vulnerabilidades en Exim. Exim es un popular servicio de mensajería usado por los sistemas en base Unix, como Linux. En caso de que el protocolo SMTP de acceso sea vulnerable el atacante podría acceder a este. Reportadas como vulnerabilidades comunes en la NVD con los nombres CVE-2010-4344 y CVE-2010-4345.
smtp-vuln-cve2011-1720	Búsqueda de vulnerabilidades en los mecanismos Cyrus SASL de autenticación del protocolo SMTP. En caso de darse, el atacante podría ejecutar un ataque DoS (Denegación de servicio en español, dejar la víctima inoperativa) o ejecutar código a distancia. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2011-1720.
smtp-vuln-cve2011-1764	Búsqueda de vulnerabilidades en el protocolo de autenticación de Exim. Exim es un popular servicio de mensajería usado por los sistemas en base Unix, como Linux. En caso de que el protocolo SMTP de acceso sea vulnerable el atacante podría acceder a

	este. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2011-1764.
ssl-ccs-injection	Búsqueda de la vulnerabilidad CSS ijnection. Esta vulnerabilidad permite al atacante inyectar código CSS con una gran variedad de diferentes objetivos. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-0224.
ssl-dh-params	Búsqueda de vulnerabilidades durante el intercambio de claves para encriptar al inicio de una comunicación (protocolo Diffie-Hellman).
ssl-heartbleed	Búsqueda de la vulnerabilidad Heartbleed de OpenSSL. Open SSL es un paquete de herramientas usado para encriptar las comunicaciones a través de la red. Esta vulnerabilidad, solo presente en la versión 1.0.1f de OpenSSL, permite a un atacante acceder a la memoria de la víctima, sea esta cliente o servidor, y obtener información como contraseñas, entre otras cosas. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2014-0160.
ssl-known-key	Búsqueda de vulnerabilidades en el protocolo SSL, un protocolo muy usado para encriptar las comunicaciones. En caso de encontrar alguna, el atacante podría tener mucha facilidad para leer la información del navegador, como las cookies, contraseñas, correos, etc.
ssl-poodle	Búsqueda de vulnerabilidades en el protocolo SSLv3, un protocolo muy usado para encriptar las comunicaciones. En caso de encontrar alguna, el atacante podría tener mucha facilidad para leer la información del navegador, como las cookies, contraseñas, correos, etc.

sslv2-drown	Testeo de las vulnerabilidades comunes reportadas en la NVD con los nombres CVE-2015-3197, CVE-2016-0703 y CVE-2016-0800, relacionadas con el protocolo SSLv2, un protocolo muy usado para encriptar las comunicaciones. En caso de encontrar alguna, el atacante podría tener mucha facilidad para leer la información del navegador, como las cookies, contraseñas, correos, etc..
supermicro-ipmi-conf	Búsqueda de vulnerabilidades en controladores Supermicro Onboard IPMI. Los softwares IPMI ("Intelligent Platform Management Interface") son unos softwares muy populares usados para la creación y monitorización de sistemas de computación autónomos. Reportado como vulnerabilidad común en la NVD con el nombre CVE-2016-9244.
tls-ticketbleed	Búsqueda de las vulnerabilidades reportadas como vulnerabilidades comunes en la NVD con el nombre CVE-2016-9244.
wdb-version	Búsqueda de vulnerabilidades usando la depuración en sistemas operativos VxWorks. El proceso de debugging o depuración consiste en la localización de errores y cosas fuera de lo normal en la aplicación o web, lo que podría dar al atacante información muy valiosa.

COMANDOS DEL ESCANEO

Así pues, a continuación, se dejan los dos comandos Nmap usados para realizar los dos escaneos, el TCP y el UDP, usando la Ip de la página web de la escuela etseib.upc.edu 147.83.2.193 como ejemplo (figuras 3 y 4):

- nmap -sS -vv -f -d9 --script vuln 147.83.2.193

- nmap -sU -vv -f -d9 --script vuln 147.83.2.193

```
bolu@bolu-VirtualBox:~$ sudo -s
[sudo] contraseña para bolu:
root@bolu-VirtualBox:~# dig +short etseib.upc.edu
147.83.2.193
root@bolu-VirtualBox:~# nmap -sS -vv -f -d9 --script vuln 147.83.2.193

Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-05 13:24 CEST
Fetchfile found /usr/bin/../share/nmap/nmap-services
```

Fig 3.1.3.1 Ejecución escaneo TCP SYN descrito con Nmap

```
bolu@bolu-VirtualBox:~$ sudo -s
[sudo] contraseña para bolu:
root@bolu-VirtualBox:~# nmap -sU -vv -f -d9 --script vuln 147.83.2.193

Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-05 13:39 CEST
Fetchfile found /usr/bin/../share/nmap/nmap-services
```

Fig 3.1.3.2 Ejecución escaneo UDP descrito con Nmap

El resultado del primer escaneo es de 21.896.214 líneas escritas en el terminal, mientras que del segundo es de 1.505.972 líneas. Hay que mencionar que en la Ip de la página web de la escuela solo hay 3 puertos UDP que respondan y 2 puertos TCP abiertos (figuras 5 y 6). El número de líneas que resultantes del escaneo aumenta mucho más si encuentra más puertos y servicios abiertos.

```
root@bolu-VirtualBox:~# nmap -sS -f 147.83.2.193

Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-05 13:51 CEST
Nmap scan report for labson.upc.edu (147.83.2.193)
Host is up (0.020s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 20.79 seconds
```

Fig 3.1.3.4 Escaneo simple que muestra los puertos TCP

Para filtrar tal cantidad de líneas se ha elaborado big_loop.py, un programa que

extrae de un fichero txt una dirección Ip y una dirección de correo electrónico. Este programa tiene como objetivo principal rastrear entre la salida devuelta por Nmap la información relativa a los puertos abiertos, los scripts utilizados durante el escaneo y la conclusión, es decir si el objetivo es seguro o no respecto a la vulnerabilidad concreta que se busca.

3.2. BIG_LOOP.PY

Todos los scripts descritos y mencionados en este apartado y los venideros están disponibles en el anexo.

Big_loop.py es un código Python de unas 350 líneas con multitud de funciones y bucles que inicia automáticamente los dos escáneres de puertos especificados en el apartado 3.1.3 y el posterior análisis de sus resultados. Estos son realizados por medio de la función getstatusoutput del módulo subprocess, quien permite la ejecución directa de comandos en el terminal y capturar su salida como una lista con un elemento por cada línea de repuesta. El comando para el escaneo TCP SYN es el siguiente:

informe_rebut = getstatusoutput('nmap -sS -v -f -d9 --script vuln '+ip)[1]

Como bien se ha mencionado anteriormente, la obtención de los dos parámetros necesarios para el funcionamiento del código, llamados *ip* y *receptor* se hace leyéndolos de un fichero de extensión .txt llamado ip.txt, situado en la ruta de directorios /tfg/app/static/ip.txt, dónde deben haber estado escritos previamente. Una vez finalizado el escaneado y creada la variable informe_rebut, esta será transformada en un primer bucle centrado en localizar los desgloses de las respuestas de los puertos abiertos para ver cuáles son estos puertos, que protocolos de comunicación usan y que scripts ha usado Nmap. Generadas estas nuevas variables, pasan directamente por un segundo bucle, esta vez, centrado en localizar y aislar los nombres de los scripts, buscar su definición en una base de datos de elaboración propia para saber con qué fines se ha aplicado el script y cuál es la conclusión del escaneado sobre él.

La localización de las líneas que hablan sobre puertos se hace a través de la función mira_si_es_port, la cual identifica si la línea corresponde al inicio de la salida de la repuesta de un puerto y que, por lo tanto, las líneas que le prosiguen son de interés (figura 3.2.1).

```
# funcion que identifica si la linia dada corresponde al inicio de la respuesta
del puerto

def mira_si_es_port(string,protocol):
        try:
                resu1 = string.split(' ')
                return resu1[0].split('/')[1] == protocol
        except:
                return False
```

Fig 3.2.1 Función mira_si_es_port, parte de big_loop.py

3.2.1. BASE DE DATOS SQL

Con tal de relacionar las palabras clave de la salida de Nmap con sus explicaciones y interpretaciones, se materializó la base de datos SQL 2db4.db (figura 3.2.1.1), situada en la ruta de directorios /tfg/app/static/2db3.db, creada y administrada con la librería Python sqlite3. Se usó esta librería por la gran facilidad que presenta tanto en el momento de conectar con la base de datos como en el momento de hacer selecciones y extracciones en lenguaje SQL si ya estás trabajando en un código Python. La base posee una única tabla llamada *tab1* con tres campos, *ARCHIVO, DEFINICIÓN* y *LINK*, que contienen los nombres de los archivos de la categoría vuln de Nmap relacionados con sus definiciones y objetivos. Además, también hay disponibles la rutas http en el campo LINK que llevan directas a la web de Nmap, para conocer toda la información posible sobre cada script en concreto.

```
bolu@bolu-VirtualBox:~/Escritorio/experiment3/app/scripts$ sqlite3 2db4.db
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite> .tables
tab1
sqlite> .schema tab1
CREATE TABLE tab1 (ARCHIVO TEXT PRIMARY KEY, DEFINICION TEXT, LINK TEXT);
sqlite>
```

Fig 3.2.1.1 Base de datos 2db4.db vista desde sqlite3

Una vez se ha terminado el filtrado del segundo bucle y la extracción de información de la base de datos SQL, esto da como resultado otra lista diferente, la cual posee los puertos abiertos, los scripts que han sido necesarios en cada uno, sus explicaciones correspondientes y sus respuestas. Llegados a este punto, hay que organizar toda esta información para facilitar su lectura y comprensión. Se eligió escribirla en un fichero de formato PDF que hiciera viable su posterior envío y llegada cómoda al usuario.

3.2.2. REPORTLAB Y EL BUCLE QUE GENERA EL PDF

La generación del PDF va a llevarse a cabo mediante la librería Python ReportLab, en un bucle dentro de otro bucle que además son capaces de llamar a una función recursiva llamada come_linia cuando se dan las condiciones adecuadas. Esto se debe a que la generación de un documento PDF con Python se hace creando una especie de imagen, lo que obliga a tener controlados el pixel de la coordenada y, el pixel de la coordenada x (en un sistema de ejes de base natural con origen en la esquina inferior izquierda) además del número de página que se está generando. La creación de este tipo de imágenes se materializa mediante los objetos de la clase canvas, disponibles en el módulo Canvas de la librería ReportLab. Se ha elegido este método porque PDF es el formato de documento más extendido y internacional del mundo. Además, con ReportLab se puede generar uno directamente desde Python, representando la información de manera clara y ordenada.

Estos bucles también llaman a la función cabezera (figura 3.2.2.1), llamada cuando se inicia un report nuevo y que se encarga de escribir el formato de inicio estandarizado por el programa, con la dirección Ip que se ha escaneado, las horas exactas de inicio y finalización de escaneo y un breve listado de los puertos que se han encontrado abiertos.

```
def cabezera(m,v,ip):
        m.drawString(50,750,'Informe script de búsqueda de vulnerabilidades:')
        m.drawString(50,730,'IP4: '+str(ip))
        m.drawString(50,710,'Fecha: '+str(time.strftime('%d:%m:%y')))
        m.drawString(50,690,'Hora inicio: '+str(hora_inicio))
        m.drawString(50,670,'Hora final: '+str(time.strftime('%H:%M:%S')))
        m.drawString(50,650,'ESTADO GENERAL DE LOS PUERTOS ABIERTOS')
        m.drawString(50,630,'PORT     STATE SERVICE')
        pix_act=610
        for z in v:
                m.drawString(50,pix_act,z)
                pix_act=pix_act-20
        return pix_act
```

Fig 3.2.2.1 Función cabezera, parte de big_loop.py

3.2.3. SMPTLib Y EL ENVÍO DE CORREOS

Con tal de automatizar el envío de correos electrónicos que sirven tanto de aviso de que se ha iniciado el proceso de escaneo a la dirección Ip especificada, como de confirmación de que este se ha finalizado y medio para enviar el informe generado, se han usado las librerías SMTPLib, email y ssl. Con funciones disponibles en la

librería email, se genera el correo y se adjunta, si es necesario, el documento PDF con el informe sobre las vulnerabilidades. Con funciones disponibles en las librerías ssl y SMPTLib, se crea la conexión a un puerto que usará el protocolo SMPT (soportado por TCP) para conectar con la cuenta gmail especificada como emisor y realizar las acciones correspondientes (figura 3.2.3.1).

```
#datos correo

subject = "Inicio escaneado con NMAPInterpreter"
body = "A las "+str(hora_inicio)+" se ha iniciado el proceso de escaneo a "+str(ip)
+". Pronto recibirá un correo con el informe solicitado.\n\nGracias por usar
NMAPInterpreter"
sender_email = "bolumar.barrera@gmail.com"
receiver_email = receptor
password = [redacted]

# Creación correo

message = MIMEMultipart()
message["From"] = sender_email
message["To"] = receiver_email
message["Subject"] = subject
message.attach(MIMEText(body, "plain"))
text = message.as_string()

# Login gmail i envio correo

context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(sender_email, receiver_email, text)
```

Fig 3.2.3.1 Código para genera y enviar un correo, parte de big_loop.py

Big_loop.py es un programa que realiza todas las funciones descritas anteriormente de forma automática, solo es necesario ejecutarlo desde el terminal de Linux o desde el cmd en la ruta de directorios correcta si se trabaja con sistemas operativos Windows. Con tal de facilitar el uso por parte de usuarios que no llegan a aventurarse a escribir comandos en el terminal, se ha montado un árbol de directorios para poder establecer un aplicativo que use Flask (popular librería de Python para hacer aplicaciones web y establecer servidores) como servidor y que se encargue de ejecutar los comandos en nombre del usuario, proporcionándole a este una interfaz cómoda y entendible, en la que no pueda perderse, para que la posibilidad de ejecutar un escaneo de este tipo y la correcta comprensión de sus resultados pueda llegar hasta los usuarios más inexpertos. Todo esto dio lugar a VULNInterpreter (figura 3.3.1).

3.3. VULNInterpreter

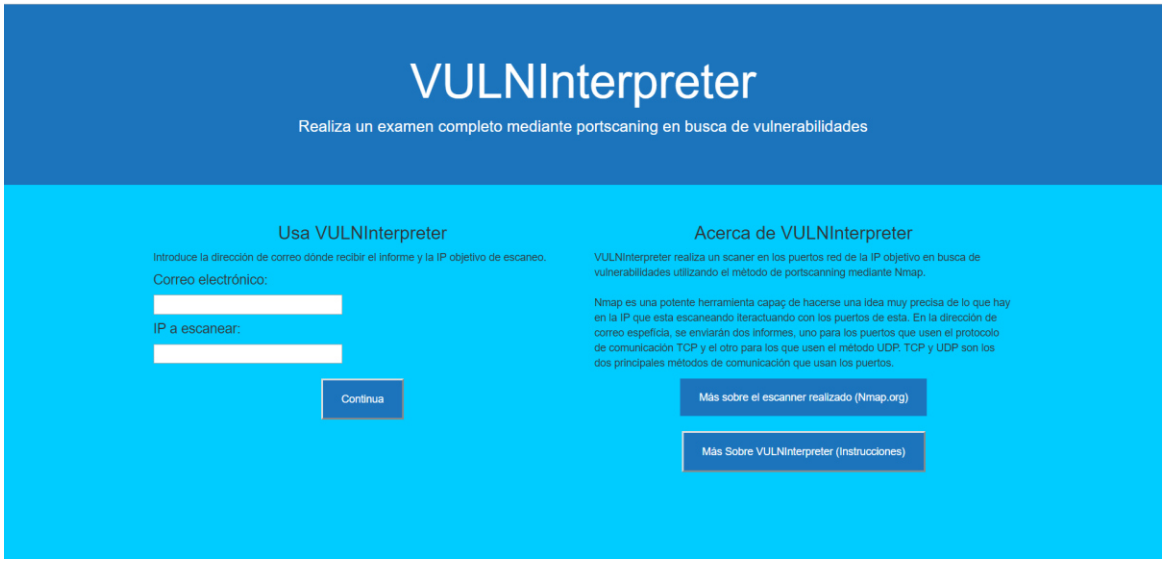


Fig 3.3.1 VULNInterpreter visto desde portátil

Fig 3.3.2 VULNInterpreter visto desde smartphone

Las imágenes anteriores (figuras 3.3.1 i 3.3.2) muestran la interfaz principal de VULNInterpreter, la aplicación creada con el objetivo de facilitar la interacción entre el usuario y big_loop.py. El servidor que soporta esta aplicación es Flask, y el fichero que contiene las acciones correspondientes a las distintas rutas del servidor es views.py (figura 3.3.3).

```
@app.route('/')
@app.route('/index')
def index():
    return render_template('interface.html')

@app.route('/inicia_prova6', methods=['POST'])
def inicia_prova6():
    text=open("app/scripts/ip.txt",'w')
    IP=request.form["IP"]
    correo=request.form["correo"]
    text.write(str(IP)+'\n')
    text.write(str(correo))
    text.close()
    try:
        if socket.inet_aton(IP):
            if re.match('^[(a-z0-9\_\-\.)]+@[(a-z0-9\_\-\.)]+\.[(a-z)]{2,15}$', correo.lower()):
                return render_template('pagina_mitja.html')
            else:
                return render_template('error2.html')
        else:
            return render_template('error1.html')
    except:
        return render_template('IPinvalida.html')

@app.route('/escaneo', methods=['POST'])
def escaneo():
        comando="./hola.sh"
        lklk=shlex.split(comando)
        subprocess.call(lklk)
        return render_template('interface.html')
```

Fig 3.3.3 Parte de views.py

En la imagen anterior, donde se ve buena parte del fichero views.py, se observa que, cuando un usuario accede al aplicativo, el servidor devuelve el fichero interface.html, fichero que tiene dos campos a rellenar, entre otras cosas. En estos campos se pide la introducción de la Ip que se pretende escanear y la dirección de correo electrónico dónde se quiere recibir la conformación de que se ha iniciado el escaneo y el informe de vulnerabilidades cuando este termine. Lo primero que se hace cuando se recibe respuesta sobre estos campos (ruta 'inicia_prova6') mediante el método POST, es actualizar el archivo ip.txt, anteriormente descrito en el inicio del apartado 3.2.

Después comprobarse de que la Ip es válida, que se puede conectar con ella y que la dirección de correo también es correcta, usando las funciones match y inet_aton de las librerías re y socket, se devuelve al usuario pagina_mitja.html (figura 3.3.4), quien contiene dos botones, 'Visualización en pantalla' y 'Envío directo a correo' con las rutas '/escaneo' y '/escaneo2 enlazadas. Dichas rutas se encargan de ejecutar los ficheros hola.sh y hola2.sh.

En caso de que la dirección Ip o la de correo electrónico introducidas por el usuario no sean válidas, se redirecciona al usuario a páginas dónde se informa de ello.

Análogos son los sucesos cuando la Ip es válida pero no se puede conectar con ella (figura 3.3.5).

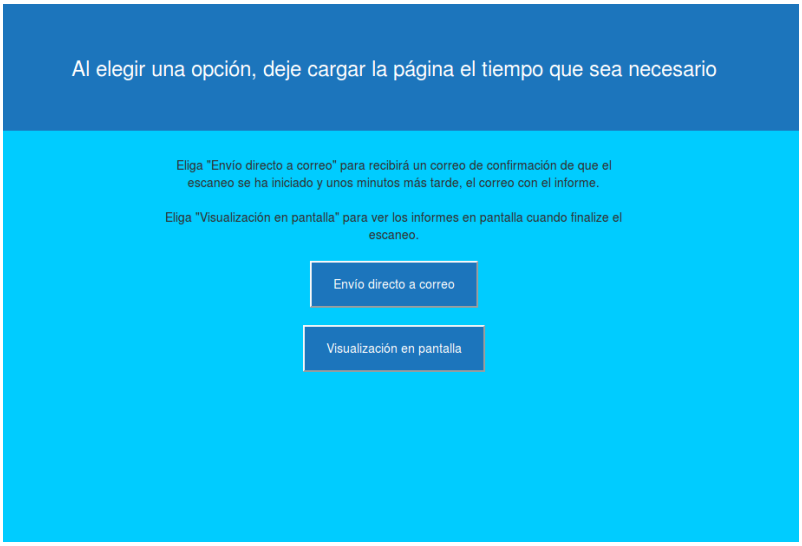


Fig 3.3.4 pagina_mitja.html

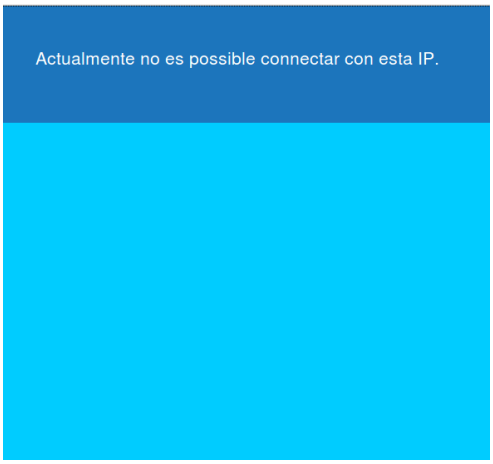


Fig 3.3.5 Pagina de redirección cuando no se puede conectar con la Ip

Hola.sh y hola2.sh (figura 3.3.6) son pequeños programas bash, es decir, ficheros que contienen comandos directamente interpretables por el terminal y que luego ejecutarán en orden. Este se encarga de guiar al servidor Flask hasta el directorio adecuado para ejecutar big_loop.py y de indicarle el protocolo que debe usar. Posteriormente, da la orden de ejecutar el código y se deja que este haga el resto del servicio. Se tuvo que recurrir a esta medida para conseguir que big_loop.py fuera capaz de conectar correctamente con la base de datos 2db4.db.

```
#!/bin/bash

cd app
cd scripts
python3 big_loop.py tcp
python3 big_loop.py udp
```

Fig 3.3.6 Fichero hola.sh

En la imagen anterior también puede observarse que se ejecuta big_loop.py dos veces, una para que realice el escaneado usando el protocolo TCP y otra para el protocolo UDP. La diferencia entre hola.sh y hola2.sh recae en que uno ejecuta la versión completa de big_loop.py, la cual mantiene “feedback” con el usuario a través del envío de correos electrónicos, y el otro una versión que simplemente genera los documentos PDF y los muestra al terminar en la página final.html.

La primera modalidad está pensada para que el usuario pueda dar la orden de iniciar el escaneo y despreocuparse luego. Los exámenes de puertos no son instantáneos y su duración va fuertemente relacionada con la cantidad de puertos abiertos en la Ip que se está examinando. Por eso se decidió hacer viable esta opción, para que el sujeto que use el aplicativo pueda recibir correos para saber que sus órdenes están siendo ejecutadas y con los informes resultantes (figura 3.3.7).

La segunda modalidad no envía ningún correo al usuario y requiere que este no cierre la aplicación para poder ver y descargar los informes, dado que estos serán cargados en la página final.html al terminar (figura 3.3.8).

Fig. 3.3.7 Correo electrónico final enviado al usuario de la primera modalidad

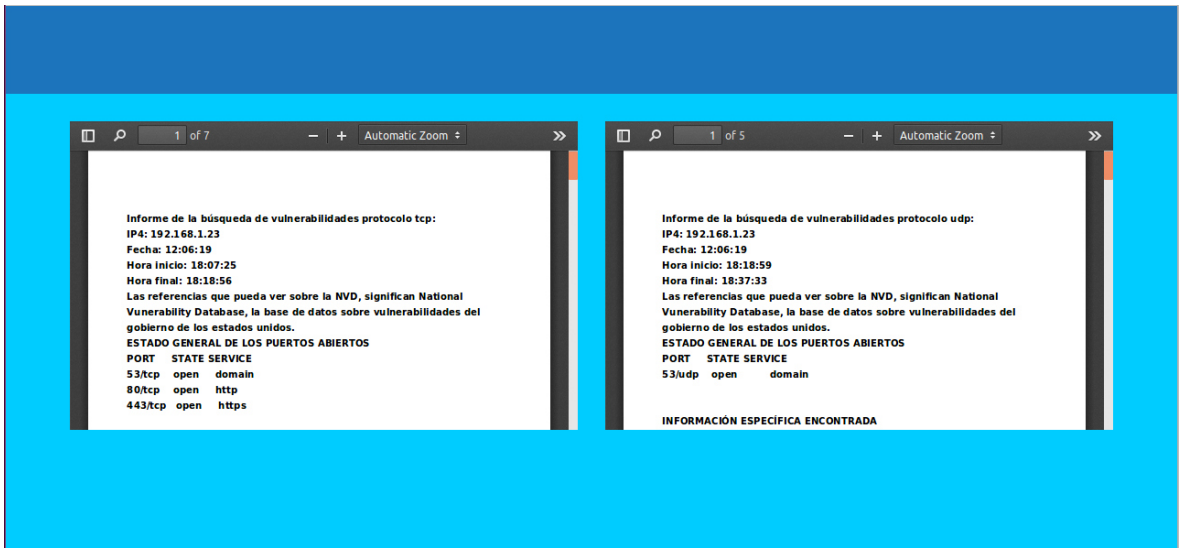


Fig. 3.3.8 Página final donde el usuario de la segunda modalidad ve los informes

La estructura de directorios y ficheros necesaria en el servidor para que funcione el aplicativo es la presentada a continuación, dónde los nombres de los directorios son representados con letra estilo negrita y los nombres de los archivos van precedidos de un punto en negro.

1. Tfg:

- Hola.sh
- Hola2.sh
- Run.py

1.1 App:

- __init__.py
- Views.py

1.1.1 Static:

- Big_loop5.py
- Big_loop2.py
- Ip.txt
- 2db4.db

- Report_tcp.pdf
- Reportudp.pdf
- Vera.ttf
- VerBD.ttf

1.1.2 Templates:

- Error1.html
- Error2.html
- Final.html
- IPinvalida.html
- Pagina_mitja.html
- Interface.html

4. PRUEBAS Y EXPERIMENTACIÓN

Con tal de verificar el correcto funcionamiento del sistema en su conjunto, se ha elaborado un plan de experimentación mediante el cual se pretende testear el funcionamiento de aplicativo en distintos ambientes de trabajo. El estudio se divide en dos grandes apartados, que son global y local, dependiendo de si la Ip objetivo se encuentra en la misma red privada o no.

Se eligieron varios objetivos para cada apartado y se midieron los siguientes parámetros durante el escaneo para ambos protocolos:

- Nº de puertos abiertos.
- Nº de vulnerabilidades localizadas.
- Tiempo de duración del escaneado.
- Nº de líneas devueltas por el examen Nmap (solo para el examen TCP).

Los resultados pueden visualizarse en las siguientes tablas:

Local

Objetivo	Nº de puertos TCP	Vulnerabilidades TCP	Nº de puertos UDP	Vulnerabilidades UDP	Tiempo de Escaneado TCP (minuto:segundo)	Tiempo de Escaneado UDP (minuto:segundo)	Lineas devueltas por Nmap en el escaneo TCP
Smartphone Apple	0	0	0	0	4:49	6:18	69.508
Smartphone Huawei	0	0	0	0	4:27	9:13	74.659
Portátil Linux	0	0	0	0	7:45	8:02	383.567
Portátil Lenovo	3	5	1	0	11:31	17:34	24.100.897
Router 1	1	2	1	0	0:32	3:03	21.130.477
Router 2	1	4	1	0	0:58	4:26	20.589.242
Router 3	3	1	1	0	1:04	2:34	24.865.403

Global

Objetivo	Nº de puertos TCP	Vulnerabilidades TCP	Nº de puertos UDP	Vulnerabilidades UDP	Tiempo de Escaneado TCP (minuto:segundo)	Tiempo de Escaneado UDP (minuto:segundo)	Lineas devueltas por Nmap en el escaneo TCP
134.0.11.136	3	4	2	0	4:06	7:51	23.480.694
82.98.160.3	10	8	3	0	11:06	6:19	26.993.956
147.83.2.193	2	5	3	0	3:41	3:14	21.896.214

En el anexo pueden encontrarse los informes generados con el aplicativo durante la fase de experimentación considerados más interesantes y representativos.

5. CONCLUSIONES

Como bien se ha podido observar en la fase de experimentación, es muy común encontrar vulnerabilidades en aquellas direcciones Ip destinadas a dar algún tipo de servicio al exterior, como por ejemplo aplicativos o páginas web, que implican la conexión de un usuario ajeno a la red interna. Eso remarca la importancia que debería tener para los administradores o las personas a cargo de estos servicios realizar auditorías de seguridad en los puertos de acceso. En caso de no hacerlo puede que algún día sufran las consecuencias, dado que probablemente son y seguirán siendo objetivos fáciles de "crakear" además de altamente atractivos para cualquier curioso.

Los smartphones y los portátiles dan mejor resultado en estos exámenes al no tener, en su gran mayoría, servicios que impliquen la conexión de usuarios exteriores (en general los smartphones no pueden contener servidores internos accesibles desde el exterior). No obstante, esto no significa que según el uso que se les dé o las aplicaciones que en estos se instalen, estas no puedan aparecer y causar graves inconvenientes.

En referencia a aquellos equipos que están libres de vulnerabilidades, como bien se comentó en el prefacio del proyecto, eso no significa que sean 100% seguros. La ciberseguridad es un tema mucho más complejo y nunca se está completamente protegido. Se pueden tener los puertos libres de vulnerabilidades, pero eso no elimina la posibilidad de que inyecten un troyano mediante un lápiz de memoria, directo al equipo, entre muchos otros ejemplos. Además, los crackers cada día innovan y encuentran nuevas vulnerabilidades no conocidas por el público y por lo tanto no detectables.

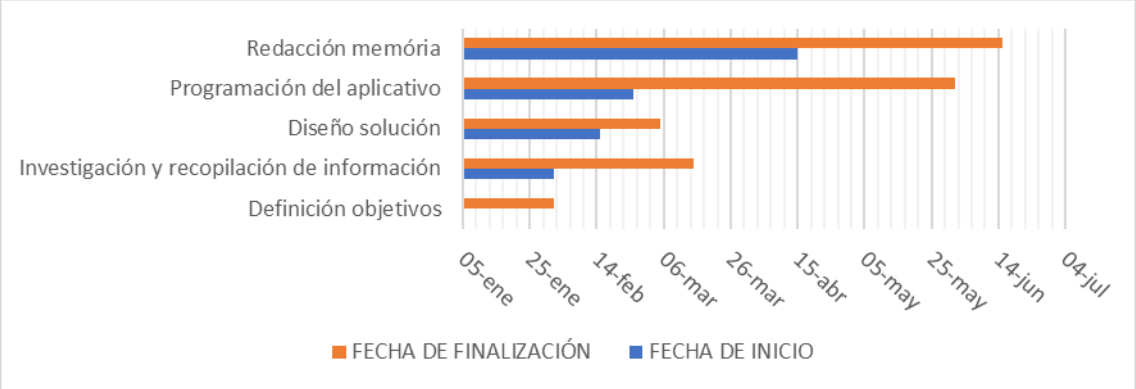
No obstante, a diario también hay hackers trabajando y colaborando con Nmap y otros grandes proyectos globales de ciberseguridad que dejan a disposición del usuario las herramientas necesarias para que este pueda hacer frente a las nuevas amenazas.

El autor del trabajo considera que las vulnerabilidades encontradas durante la fase de experimentación son suficientes para poner en entredicho el concepto de falsa ciberseguridad en el que vivimos y para garantizar que el funcionamiento del sistema VULNInterpreter es correcto. Por lo tanto los objetivos han sido cumplidos. Además, el rango de valores del campo “Líneas devueltas por Nmap en el escaneo TCP” observado, que se extiende aproximadamente desde 69.000 hasta 27.000.000, pone de manifiesto la utilidad VULNInterpreter. Nmap puede gozar de una gran popularidad entre los expertos en ciberseguridad, pero está claro que extraer las conclusiones adecuadas entre 20 millones de líneas de respuesta muy técnicas es algo fuera del alcance de la mayoría de la población.

6. COSTES Y PLANIFICACIÓN

6.1 PLANIFICACIÓN

La planificación del proyecto englobando todas las etapas, desde su concepción hasta su finalización, sigue el siguiente diagrama de Gantt:



Las 300 horas invertidas han sido distribuidas de manera bastante uniforme durante los meses de ejecución del plan. El apartado más intenso y en el que más horas se empleó en relación a los días invertidos fue “Investigación y recopilación de información”, por la complejidad del tema elegido y lo increíblemente técnica y específica que es su documentación disponible. No obstante, la “Programación del aplicativo” representa la inversión de tiempo más grande.

6.2 Costes

Gracias a que Python, Nmap y las demás herramientas y lenguajes utilizados son todos de licencia libre y gratuitos, los costes en material y elaboración de prototipos son prácticamente nulos.

Los únicos costes asociados al proyecto son las horas de trabajo de un ingeniero y el desgaste del hardware informático acumulado durante estos meses.

6.2.1 Horas del ingeniero

Según el plan de estudios de la Etseib, el valor asociado al trabajo final de grado es de 12 créditos, lo que equivale a 300 horas de trabajo. Extrapolando el precio que cobraría la empresa por la que trabaja el autor del trabajo por la implantación de procesos parecidos a sus clientes, lo que significa 30€/hora, se obtiene un precio de 9.000€.

6.2.2 Costes de hardware

El ordenador usado para desarrollar el software tiene un valor de mercado de 871€. Considerando que su vida útil es de 6 años de los cuales estuvo 5 meses dedicado a esto, esto da un coste de 60,48€.

La siguiente tabla resume las horas dedicadas a cada fase del proyecto (desglosándolas en subapartados cuando se a considerado adecuado) con los cotes asociados de hardware y personal.

Fase	Horas Invertidas	Coste de Hardware asociado (€)	Coste de Personal asociado (€)
Redacción memoria	**30**	**6,05**	**900**
Programación aplicativo	**190**	**38,3**	**5700**
big_loop.py	120	24,19	3600
Interficíes	40	8,06	1200
Estructura Servidor	20	4,03	600
Base de datos SQL	10	2,01	300
Diseño solución	**15**	**3,02**	**450**
Investigación	**65**	**13,1**	**1950**
Nmap	55	11,09	1650
Comunicaciones en la red	10	2,01	300
TOTAL	**300**	**60,48**	**9000**
COSTE TOTAL PROYECTO (€)	**9.060,48 €**		

BIBLIOGRAFÍA

Referencias bibliográficas

[1] El País

[https://elpais.com/tecnologia/2018/02/27/actualidad/1519725291_071783.html]

[2] Ministerio del interior

[http://www.interior.gob.es/documents/10180/5791067/Estudio+Cibercriminalidad+2016.pdf/456576b2-9ce8-4f3c-bbcc-ca0dbf3bb3cf]

[3] Diccionario de Hacking, Daniel Maldonado

[http://danielmaldonado.com.ar/diccionario-de-hacking/que-es-una-vulnerabilidad/]

[4] Universidad Autónoma del estado de Hdalgo

[[https://www.uaeh.edu.mx/scige/boletin/huejutla/n3/r1.html]

Bibliografía complementaria

Para la creación del aplicativo, se han usado las siguientes páginas como recurso informativo:

- https://stackoverflow.com/
- https://www.welivesecurity.com/la-es/2015/02/12/auditando-nmap-scripts-escanear-vulnerabilidades/
- https://nmap.org/

Agradecimientos

A Lluís Solano, por su paciencia y consideración.